



# **Università degli Studi di Bologna**

## **Facoltà di Ingegneria**

### *Progettazione di Applicazioni Web T*

#### **Esercitazione 1**

#### **XML e grammatiche XSD/DTD**

# Agenda

---

- Creazione di grammatiche XML a partire da una specifica dei requisiti descritta in linguaggio naturale
  - **XML Schema**
  - DTD, per esempi
- Creazione di documenti XML istanza di una grammatica

## Specifica di “Lettera”

---

- Si progetti una **grammatica XML Schema** in grado di modellare il contenuto informativo di una “**Lettera**”
- Il contenuto della Lettera è soggetto alle seguenti specifiche:
  - ogni **lettera** è caratterizzata da un mittente, una data, un destinatario, un oggetto, una forma cortese di saluto, un corpo, una chiusura, una firma
  - il corpo della lettera è costituito da almeno un paragrafo
- Si scriva (mediante l’editor di testo IDE Eclipse) inoltre un **documento XML valido per le grammatiche** al punto precedente

## Specifica di “Address List”

---

- Si progetti una **grammatica XML Schema** per la gestione di “**Address List**”
- Address List deve rispettare le seguenti specifiche:
  - ciascun **address list** contiene almeno una informazione
  - ogni informazione include: un nome, zero o più indirizzi, zero o più numeri di telefono, zero o più indirizzi email, eventualmente la nazionalità e zero o più note
  - il nome contiene: un nome proprio, zero o più secondi nomi e il cognome
  - un indirizzo è composto da: almeno una via, un indirizzo postale, eventualmente la provincia e sicuramente lo stato
- Si scriva inoltre un **documento XML valido per la grammatica** al punto precedente

## RSS 0.92 (1)

---

- Si progetti una **grammatica XML Schema per i feed Really Simple Syndication (RSS)** versione 0.92
  - si veda come **esempio di documento XML il file *RSS-0.92-example-gratefulDead.xml*** messo a disposizione come materiale di partenza dell'esercitazione
- In questo esercizio omettiamo le specifiche dettagliate di RSS 0.92 (per altro disponibili online) e ci limitiamo a descrivere elementi ed attributi che compongono un feed
- L'esempio è più complesso dei precedenti e mira a mostrare XML Schema possano essere applicati in situazioni reali nella modellazione di documenti con una struttura complessa

## RSS 0.92 (2)

---

- Ogni feed RSS ha un canale (**channel**)
- Inoltre ogni feed ha un attributo versione (**version**) il cui valore è (nel nostro esempio) 0.92
- Il canale può inoltre avere zero o più sotto-elementi scelti fra:
  - title
  - description
  - link
  - language (opzionale)
  - item (almeno un item)
  - rating (opzionale)
  - image (opzionale)
  - textInput (opzionale)
  - copyright (opzionale)
  - pubDate (opzionale)
  - lastBuildDate (opzionale)
  - docs (opzionale)
  - managingEditor (opzionale)
  - webMaster (opzionale)
  - skipHours (opzionale)
  - skipDays (opzionale)
  - cloud (opzionale)

## RSS 0.92 (3)

---

- Un'immagine (**image**) può a sua volta avere zero o più sotto-elementi scelti fra:
  - title
  - url
  - link
  - width (opzionale)
  - height (opzionale)
  - description (opzionale)
- Una **item** può avere zero o più sotto-elementi scelti fra:
  - title (opzionale)
  - link (opzionale)
  - description (opzionale)
  - source (opzionale)
  - enclosure (opzionale)
  - category (opzionale)

## RSS 0.92 (4)

---

- Sia l'elemento **source**, sia l'elemento **enclosure**, qualora siano presenti in un feed RSS, devono avere un attributo **url**
- Inoltre, l'elemento **enclosure** deve anche avere due ulteriori attributi: **length** e **type**
- L'elemento **category**, se presente in un feed RSS, ha un attributo opzionale **domain**
- L'elemento **textInput**, se presente in un feed RSS, può avere zero o più sotto-elementi scelti fra:
  - title
  - description
  - name
  - link



## RSS 0.92 (5)

---

- L'elemento **cloud**, qualora presente in un feed RSS, deve avere i seguenti attributi (tutti obbligatori):
  - domain
  - port
  - path
  - registerProcedure
  - protocol
- Infine gli elementi **skipDays** e **skipHours**, se presenti in un feed, devono avere rispettivamente almeno un sotto-elemento **day** ed un sotto-elemento **hour**

# **APPENDICE**

**(a integrazione, anche esempi basati su DTD...)**

## Da DTD a documento XML: “Ricette”

---

- Scrivere una descrizione testuale ed un documento XML valido per la grammatica DTD “Ricette”

```
<!ELEMENT recipes (recipe+, document_info)>
<!ELEMENT recipe (recipe_head, recipe_body, recipe_footer?)>
<!ELEMENT recipe_head (recipe_name, recipe_author?, meal_type)>
<!ELEMENT recipe_name (#PCDATA)>
<!ELEMENT recipe_author (#PCDATA)>
<!ELEMENT meal_type (#PCDATA)>
<!ELEMENT recipe_body (ingredients, directions)>
<!ELEMENT ingredients (ingredient+)>
<!ELEMENT ingredient (#PCDATA)>
<!ELEMENT directions (direction)+ >
<!ELEMENT direction (#PCDATA)>
<!ELEMENT recipe_footer (serving?, preparation_time?, cooking_time?)>
<!ELEMENT serving (#PCDATA)>
<!ELEMENT preparation_time (#PCDATA)>
<!ELEMENT cooking_time (#PCDATA)>
<!ELEMENT document_info (document_author, date_updated, source)>
<!ELEMENT document_author (#PCDATA)>
<!ELEMENT date_updated (#PCDATA)>
<!ELEMENT source (#PCDATA)>
```